

Choosing the model and tooling
Seth Therrien, Onshore Consulting, October 2026

Choosing the model and tooling

A one-page AI strategy and cost review

By Seth Therrien | Onshore Consulting + Deliveryboard

Your team may already be paying for a chat workspace, coding agents, and AI features inside other products. Before adding another subscription, ask what it costs to get one piece of work accepted.

> THE RULE IN ONE LINE

> Choose the simplest workflow and lowest-total-cost model that pass your acceptance tests; cap the spend before you scale.

1. Separation of duties

Keep judgment, execution, and approval distinct

Activity	Accountable person	Evidence to retain
Define the outcome	Business owner	Success criteria and baseline
Choose pattern and vendor	Technical lead; security approves data use	Chosen design and approved data path
Define good enough	Domain owner and engineering reviewer	Versioned rubric and checked answers
Implement the pilot	Engineering owner	Diff, test output, and unresolved questions
Control spend and access	Budget owner and platform administrator	Limits, permissions, and tested stop path
Accept and release	Named reviewer and release owner	Acceptance and release approval

Model is the inference capability for a task. Vendor is the organization and contract through which you access it. Tooling is the interface, coding agent, API integration, or delivery system around it. Pattern is the sequence of steps, checks, and permissions that turns a response into a result. Keep business authority outside the model.

2. Choose the workflow before the model

Write the input, expected output, and acceptance check before selecting a model. A fixed workflow follows predefined steps; an agent chooses some of its own steps and tools. More autonomy is a design choice to justify, not a maturity badge.

Use case	Starting pattern	What must be checked
Known rules and clean data	Existing software or deterministic code	Rules and reconciliation
Labels, extraction, bounded draft	One model call with defined output	Required fields and facts

Answers from company information	Retrieve permitted sources then answer	Source support and access
Predictable steps	Fixed workflow with checks between steps	Each handoff and final outcome
Repository work	Coding agent in constrained environment	Tests, scope, security, reviewer acceptance

Use retrieval for missing information, not missing accountability. For a first pilot, prefer draft-only or read-only behavior.

3. Select model and vendor by evidence

Compare the current approach with a simpler or cheaper candidate and a more capable reference. Include a non-AI option when rules are explicit. First apply non-negotiable data, contract, operational-fit, buying, and exit gates. Then assign work by observed performance and risk. A routing policy can be tested: known low-risk task to lower-cost route to validation; failed quality check to one approved escalation or human review; high-consequence task to the designated expert path.

4. Cost management

Measure cost per accepted result

> TWO NUMBERS TO CALCULATE

> Total workflow cost = allocated licenses + all model and tool charges + other operating cost + review hours x hourly rate.

> Cost per accepted result = total workflow cost / accepted results.

Keep the acceptance standard fixed. Include failed drafts and retries in the numerator. With no accepted results, report the spend and failures instead of a unit cost. A capped draft run with zero accepted items has no cost per accepted item to report, only the spend and what failed.

Illustrative monthly scenario: a lower-cost route with \$40 in model charges, 30 review hours at \$60, and \$200 operating cost totals \$2,040 for 1,000 accepted items, or \$2.04 each. A capable route with \$160 model charges, 8 review hours at \$60, and \$200 operating cost totals \$840, or \$0.84 each. The cheaper model can cost more overall.

5. Put evaluation gates and a real stop in place

An eval is a repeatable test, not an impressive demo. Define expected outputs or a scoring rubric before choosing a winner. Use representative cases, known failures, difficult inputs, and held-out cases. Record acceptance, critical errors, human correction minutes, escalation frequency, cost per accepted item, and completion time.

A budget alert is not a spending boundary. Use trusted runtime or gateway controls, reserve conservative exposure before charges, bound inputs, outputs, tools, and retries, and test the stop path with a deliberately tiny allowance.

6. Two practical walkthroughs

Support triage and reply drafts

Use a fixed workflow: categorize, retrieve permitted policy material, draft, and validate required fields and source support. Staff send replies. Test routine and complex cases separately, retain the cheaper route only where it meets the standard, and keep draft-only behavior until a separate release decision permits more authority.

A tested warehouse metrics change

Use a thinking partner to expose ambiguous definitions, then let the business owner approve grain, sources, and edge cases. A coding agent implements bounded layers from the approved spec. Require hand-checked expected values, independent reconciliations, repository checks, and reviewer acceptance.

7. Four more situations, and the sheet filled in

Most AI buying decisions look like one of these

The following illustrative situations show what enough evidence looks like before a buyer runs their own review. Load any completed sheet into the live worksheet: [Situation C: tool sprawl](/guides/ai-strategy-cost-review?example=C), [Situation D: coding agent](/guides/ai-strategy-cost-review?example=D), [Situation E: knowledge assistant](/guides/ai-strategy-cost-review?example=E), or [Situation F: document extraction](/guides/ai-strategy-cost-review?example=F).

****Situation C: tool sprawl.**** A company has multiple chat subscriptions, coding-agent seats, and AI add-ons in other products. Inventory owners, paid and active seats, recent charges, and a sampled accepted output before cancelling anything. The filled sheet chooses consolidate, downgrade, replace after testing, or cancel at renewal. Cost per accepted result is not meaningful for the inventory; report each line.

****Situation D: a coding-agent pilot.**** A lightly tested monolith needs accepted pull requests with less review time. Scope the agent to tests and bounded refactors, use its own branch, retain human merge authority, and evaluate held-out tasks with blind review. The filled sheet keeps the cheaper route only for passing task classes.

****Situation E: an internal knowledge assistant.**** Retrieve only documents the asker may see, test retrieval separately, and allow "not found." The filled sheet compares existing search, a lower-cost single-source answerer, and a capable multi-source reference. Stop on any access-control failure.

****Situation F: supplier document extraction.**** First test deterministic templates for common layouts, then a lower-cost model with a strict schema for the long tail, and use a capable reference only where reconciliation fails. The filled sheet measures correction minutes and cost per accepted document across all three routes.

When the cap fires on the first run

A draft workflow can stop at its cap with zero accepted drafts. That is the control working.

Report spend and failure classes, not a unit cost; read failures before buying more attempts. When a validator needs exact text, let code insert the fixed text and have the model write only the variable part, then prove the change on failures at zero cost.

Situation	Workflow before model	Authority outside model	Eval before pilot
C. Tool sprawl	Inventory first	Owner per line	Parallel run
D. Coding agent	Tests before refactors	Human merge	Held-out PR review
E. Knowledge assistant	Retrieval tested separately	Permission filter	Questions by class and access tests
F. Document extraction	Template route first	Reconciliation failures go to a person	Held-out documents and reconciliation

8. The printable one-page review

Complete one sheet per workflow. Use "unknown" where evidence is missing; do not turn a missing number into a saving. The same sheet is available as an interactive page at onshore.consulting/guides/ai-strategy-cost-review; it does the arithmetic in section 3 for you.

1. Outcome and ownership: workflow, owner, review period, volume, and what accepted means.
2. What we already pay for: tools, seats, licenses, metered usage, review, and commitments.
3. Baseline and alternatives: current approach, simpler candidate, capable reference, evidence, and verdict.
4. Chosen model, vendor, tooling, and pattern: exact settings, data sources, and escalation boundary.
5. Proof and release gates: eval, thresholds, allowed data and actions, approval, and fallback.
6. Approved limits and enforcement: dates, cap, task bounds, enforcer, headroom, stop, and restart owner.
7. Decision and next review: action, reason, savings, approval, next review, and rollback trigger.

9. A 30-day starter plan

Choose one workflow with a real owner and observable outputs. First gather invoices, usage, sample outputs, review time, and acceptance criteria. Next replay the same cases through candidates and prove limits and stop behavior. Then run a bounded, draft-only pilot and log corrections, escalations, and spend. At day 30, complete the review and decide to keep, simplify, downgrade, replace, or stop.

10. From review to delivery

The objective is not to buy the fewest tools or the most powerful model. It is to make the next unit of accepted work dependable and economical without losing control.

> RUN THE REVIEW ON ONE WORKFLOW

> The worksheet at onshore.consulting/guides/ai-strategy-cost-review calculates cost per accepted result for your current approach, a cheaper candidate, and a capable reference as you type. It keeps your sheet in your browser and prints to one page. Send a completed sheet

for a written read before any call. Bring a redacted example and rough spend, not credentials or customer data.

Source notes and scope

This guide extends Two Brains, One Pipeline. The review method and examples are proposed operating practices, not savings guarantees. Verify current product terms before buying. Section 7 situations are composites of common buying decisions; figures are illustrative and not reported client results.